Spectral Nets
Recurrent Nets

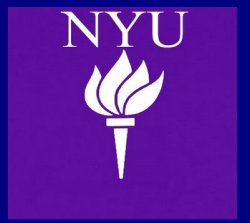
Yann Le Cun

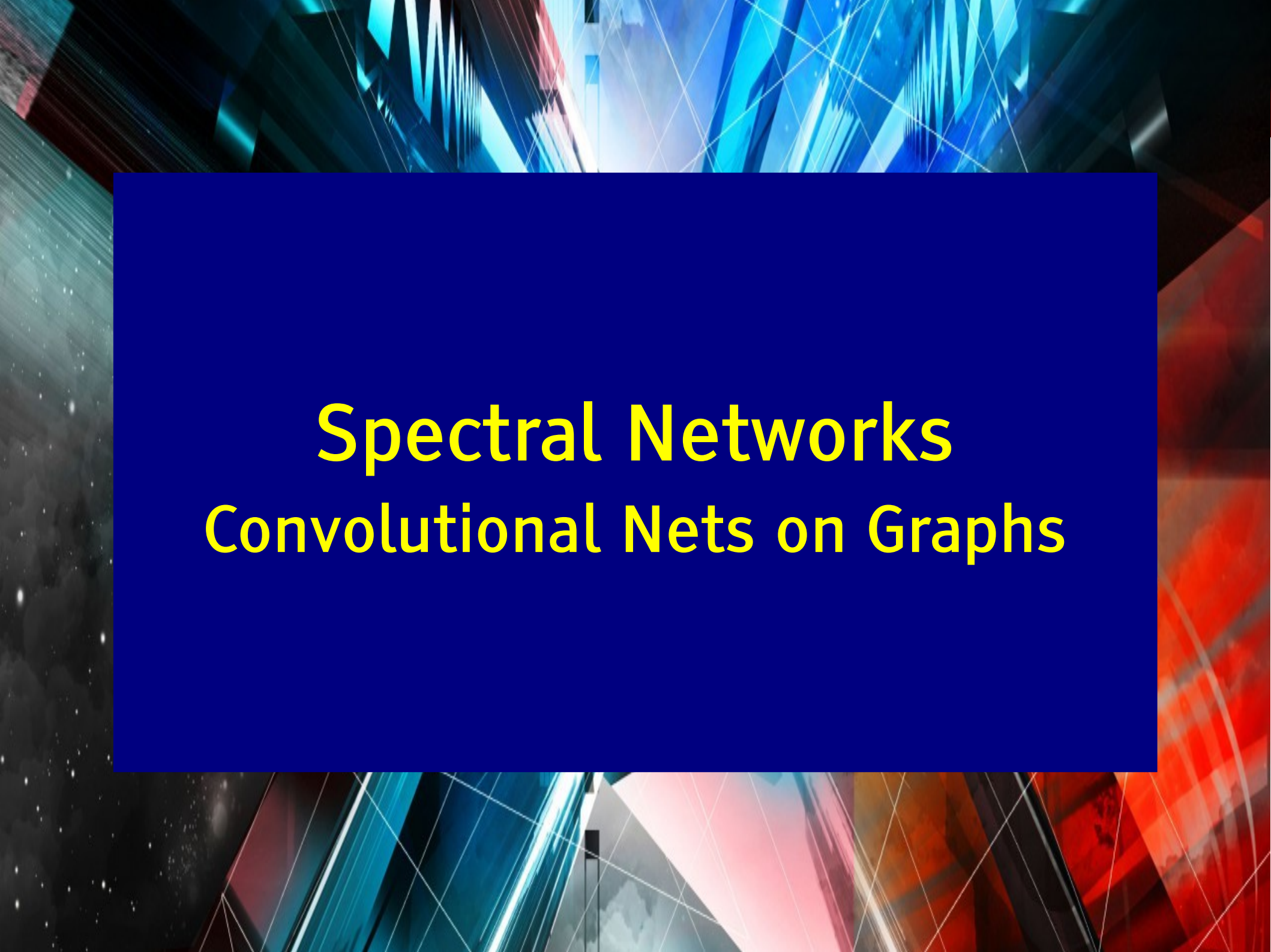
Facebook AI Research,

Center for Data Science, NYU

Courant Institute of Mathematical Sciences, NYU

http://yann.lecun.com

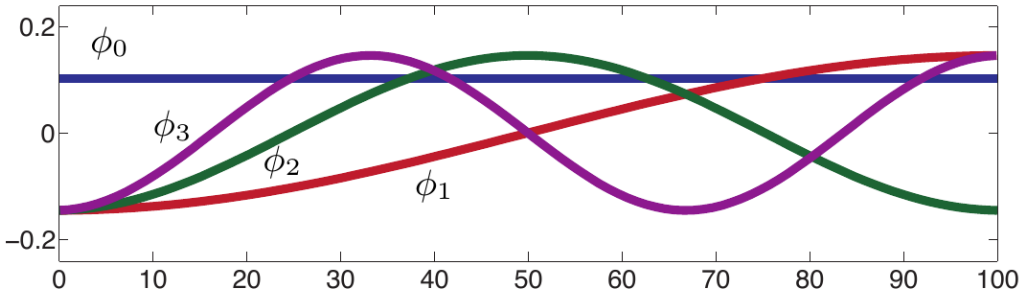



Spectral Networks
Convolutional Nets on Graphs

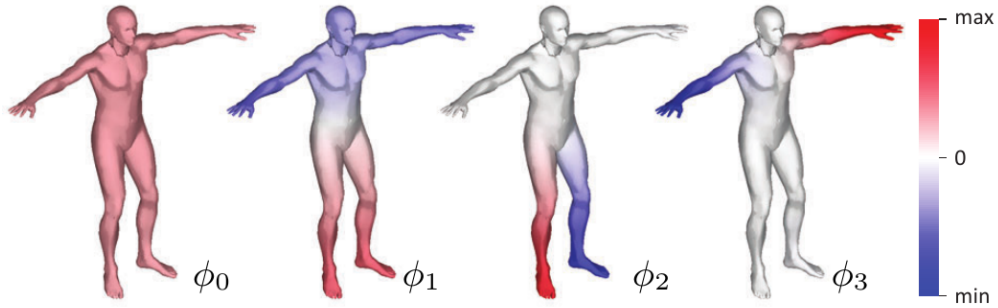
Spectral Networks: Convolutional Nets on Irregular Graphs

Y LeCun

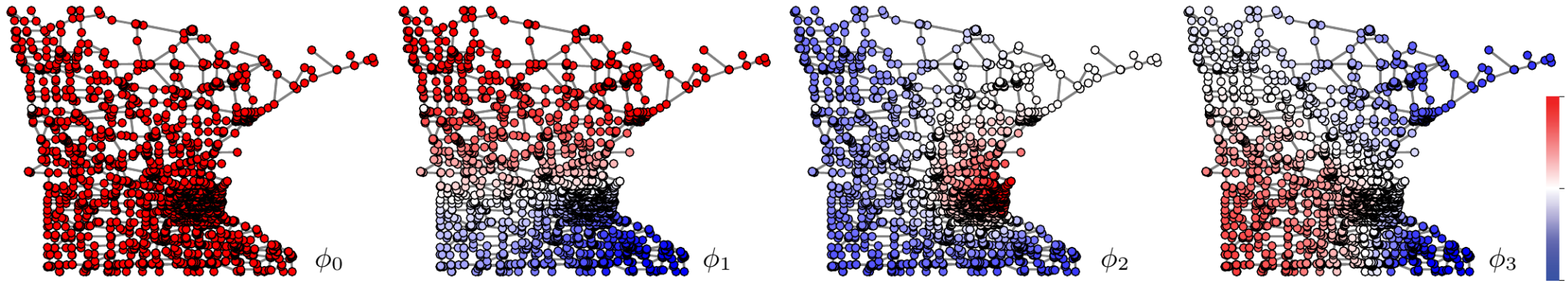
- Convolutions are diagonal operators in Fourier space
- The Fourier space is the eigenspace of the Laplacian
- We can compute graph Laplacians
- Review paper: [Bronstein et al. 2016, ArXiv:1611.08097]



Euclidean



Manifold

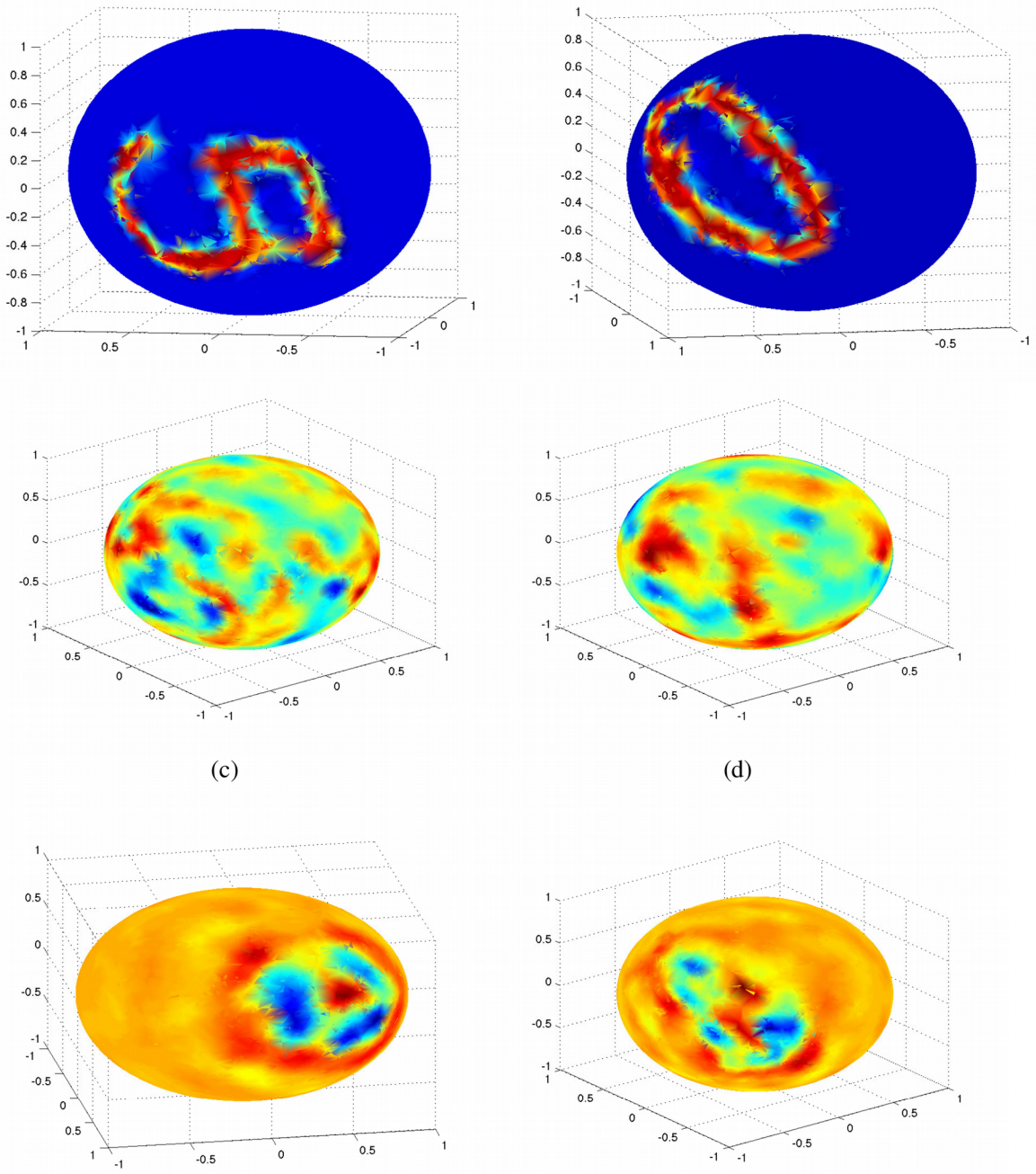


Graph

Spectral Networks: Convolutional Nets on Irregular Graphs

Y LeCun

- Compute graph Laplacian
- Compute transformation into its eigenspace
  - Fourier transform
- Learn "smooth" pointwise multipliers in Fourier space
  - Localized kernels
- Applicable to any function on graphs
  - Chemistry, text, images with holes, image on non-euclidean surfaces...
  - [Bruna et al. Arxiv:1312.6203]
  - [Henaff et al. Arxiv:1506.05163]

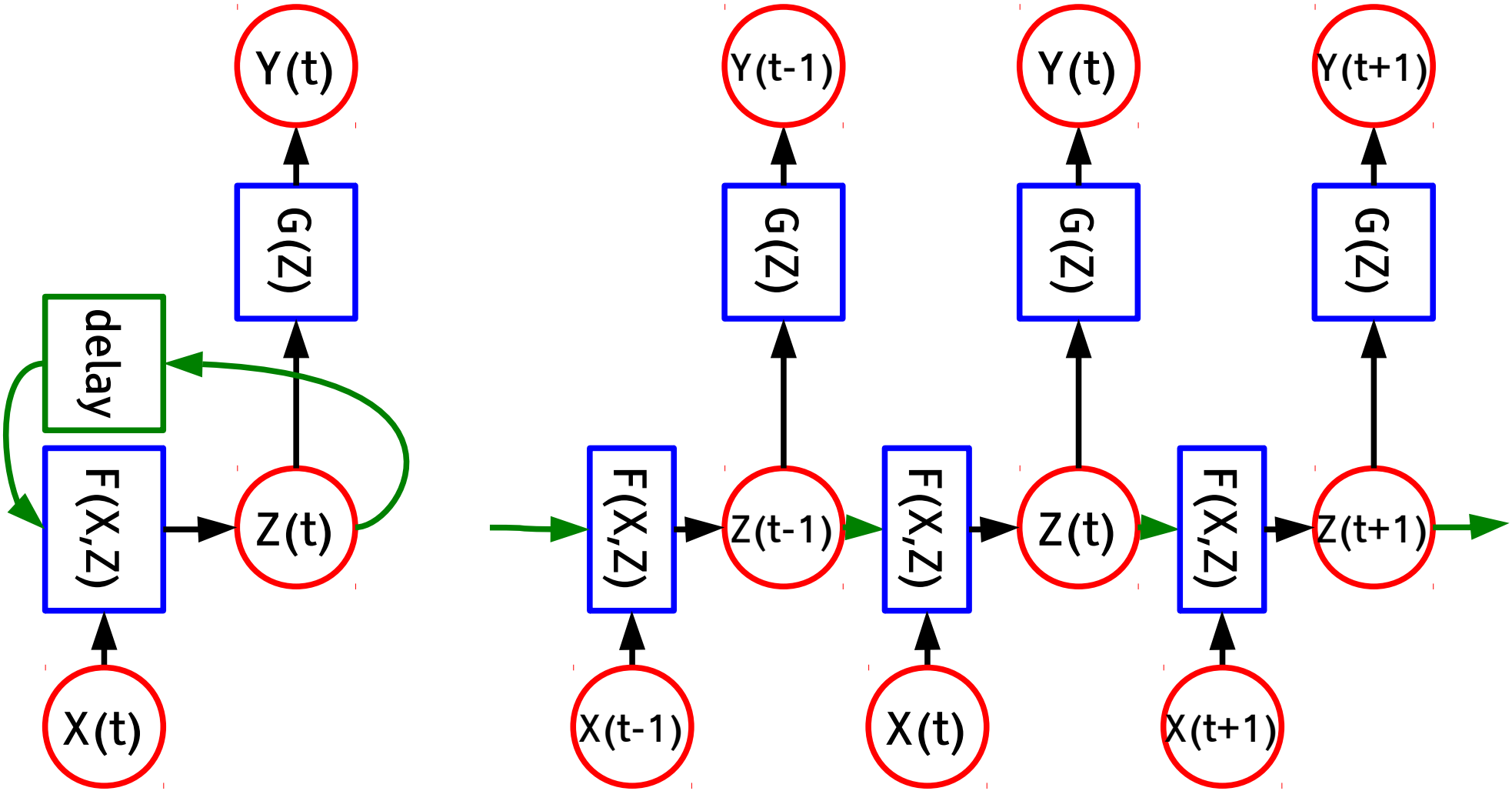


(c) (d)


Recurrent
Networks

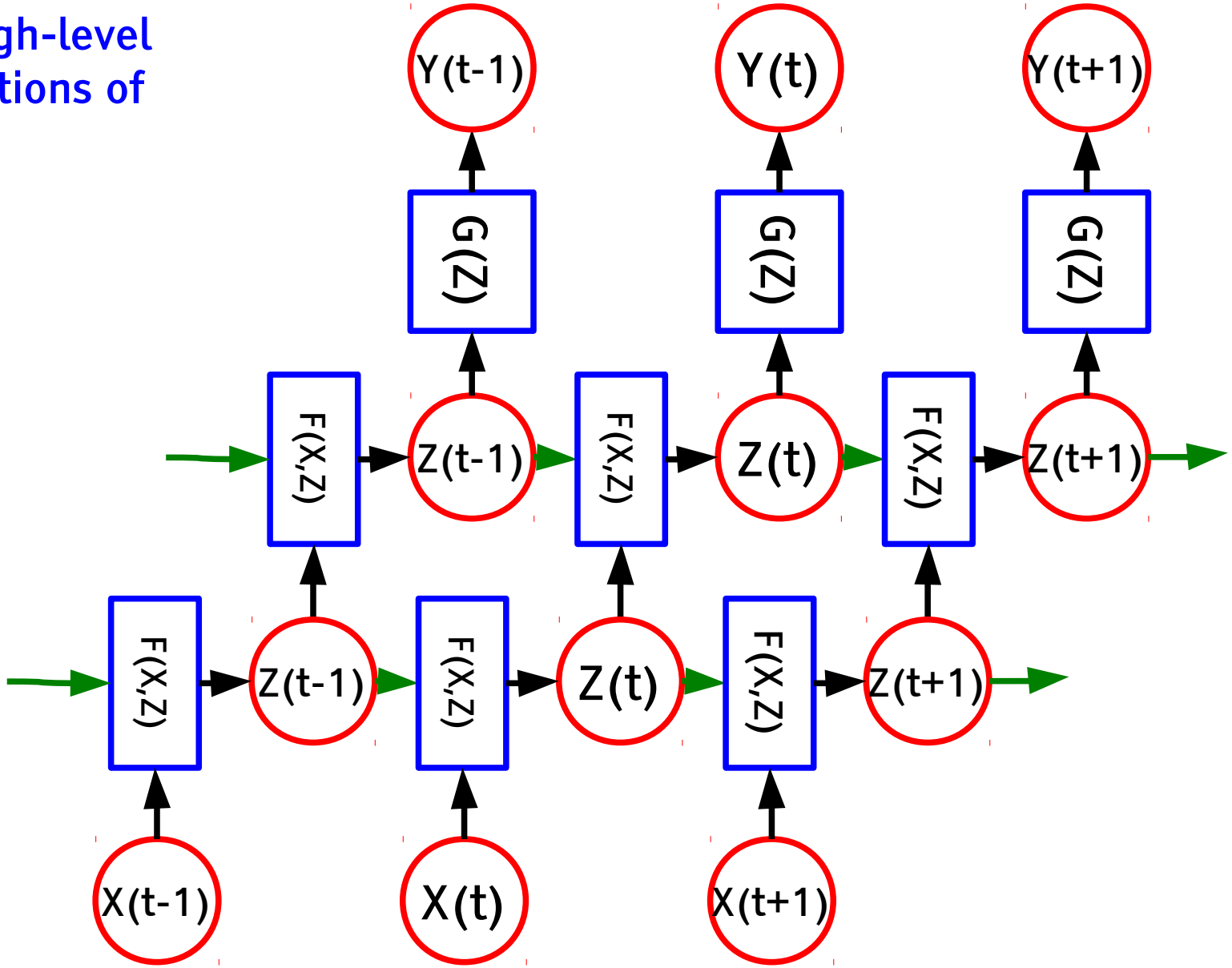
Simple Recurrent Net: Dynamical System

- The state at time t depends on the state at time t-1
- Loop unrolling: turns recurrent net into feed-forward net with shared weights
- Backprop training: “backprop through time”



RNN with "Depth"

- To learn high-level representations of sequences



Long-Term Dependencies

- The RNN gradient is a product of Jacobian matrices, each associated with a step in the forward computation. To store information robustly in a finite-dimensional state, the dynamics must be contractive [Bengio et al 1994].

$$L = L(s_T(s_{T-1}(\dots s_{t+1}(s_t, \dots))))$$

$$\frac{\partial L}{\partial s_t} = \frac{\partial L}{\partial s_T}\frac{\partial s_T}{\partial s_{T-1}} \cdots \frac{\partial s_{t+1}}{\partial s_t}$$

Storing bits robustly requires sing. values<1

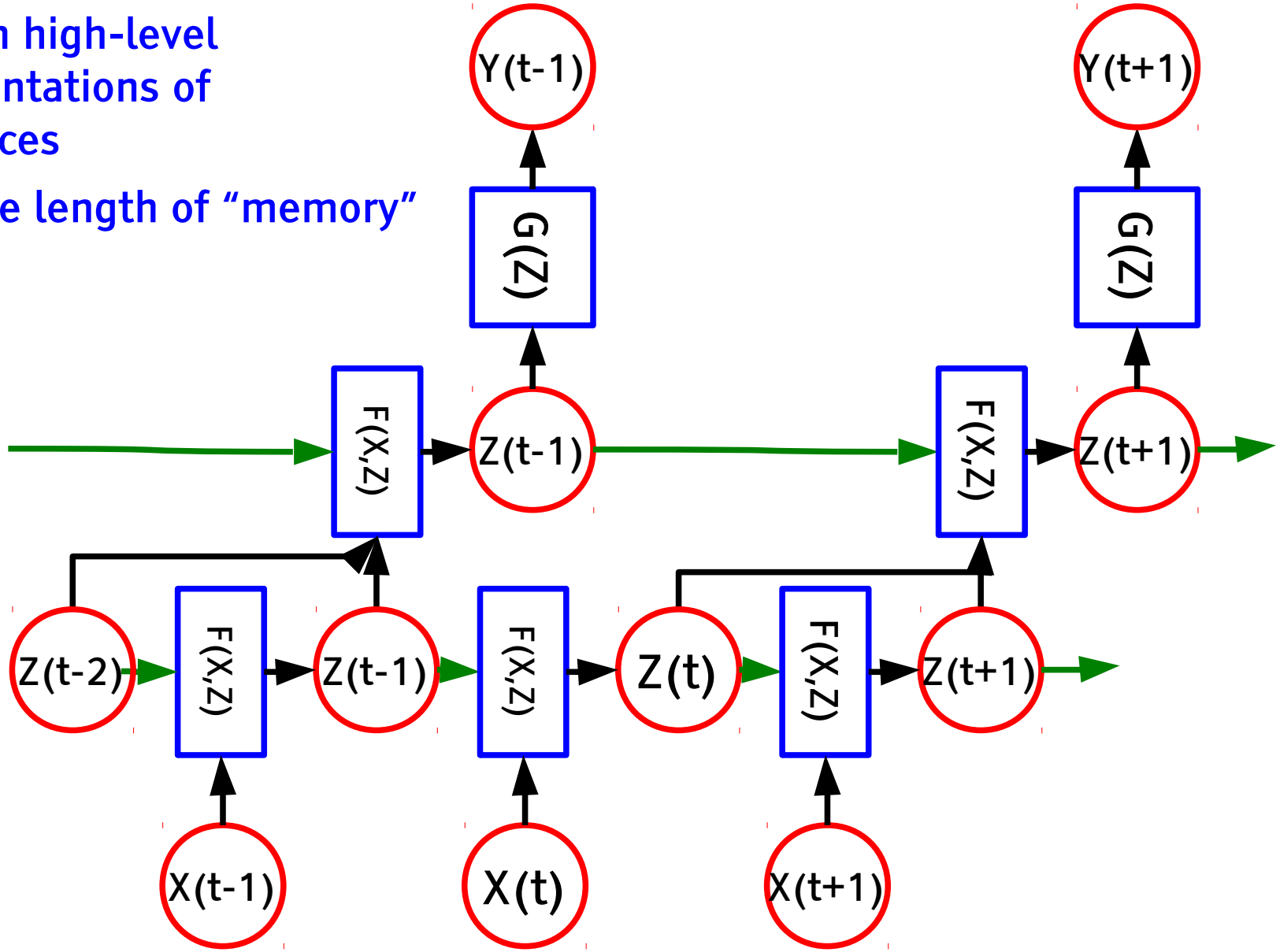
- Problems:
  - sing. values of Jacobians > 1 ⇨ *gradients explode*
  - or sing. values < 1 ⇨ *gradients shrink & vanish*
  - or random ⇨ variance grows exponentially

→ **Gradient clipping**

(Hochreiter 1991)

RNN with depth and temporal subsampling

- To learn high-level representations of sequences
- Increase length of "memory"



RNN with more memory: "Slow" RNN

- Some state variable are constrained to evolve slowly (matrix near unity)
  - Slow state used as context for faster state.
- "Learning Longer Memory in Recurrent Neural Networks",
  - Tomas Mikolov, Armand Joulin, Sumit Chopra, Michael Mathieu, Marc'Aurelio Ranzato [arxiv:1412.7753]

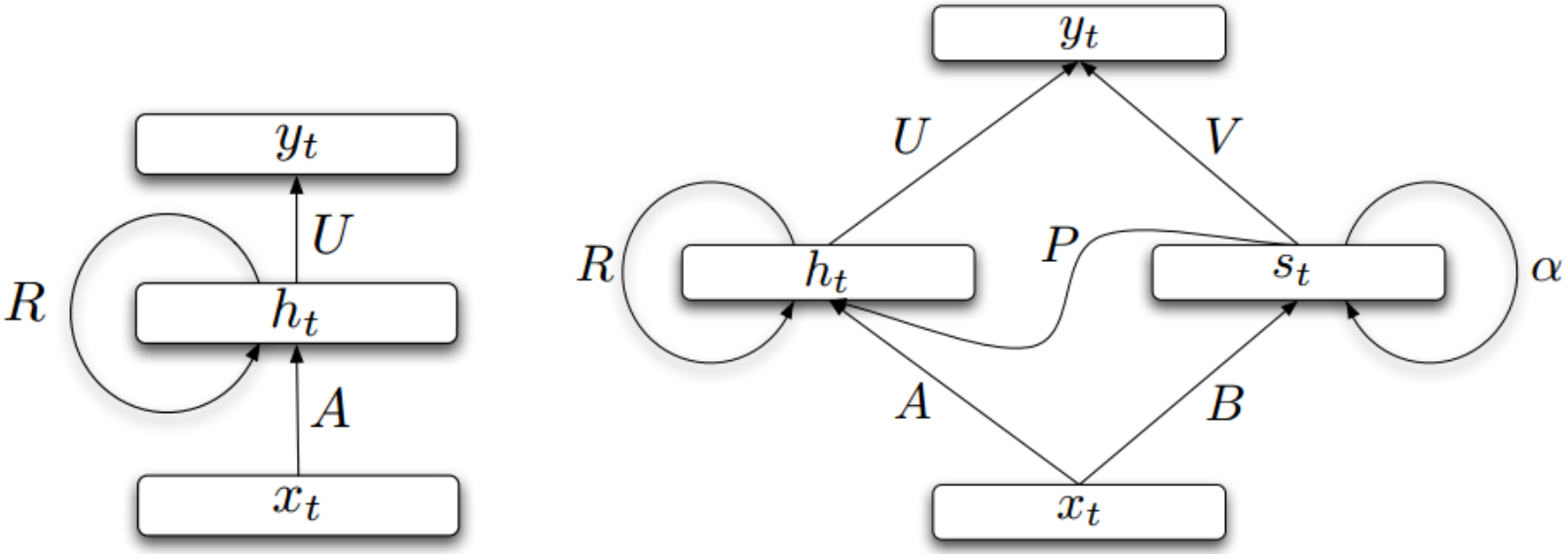
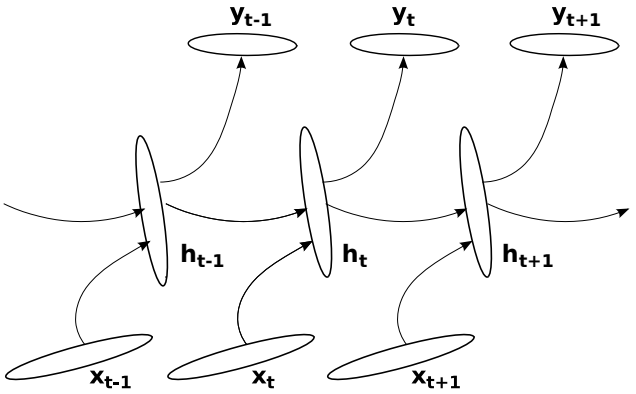


Figure 1: (a) Simple recurrent network. (b) Recurrent network with context features.

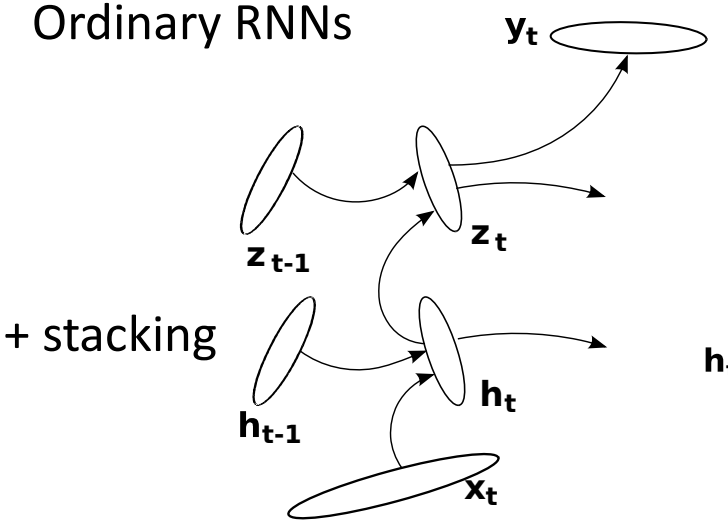
Increasing the Expressive Power of RNNs with more Depth

- ICLR 2014, *How to construct deep recurrent neural networks*

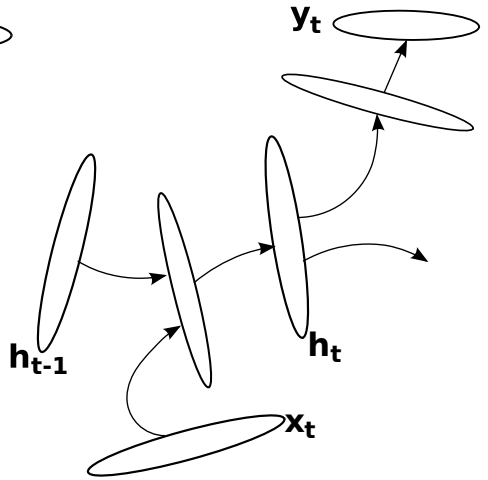


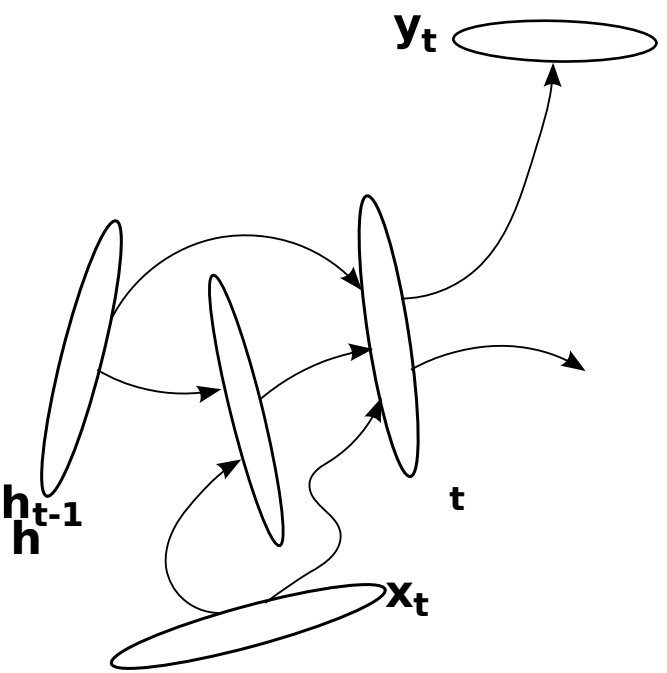
Ordinary RNNs

+ deep hid-to-out
+ deep hid-to-hid
+deep in-to-hid



+ stacking

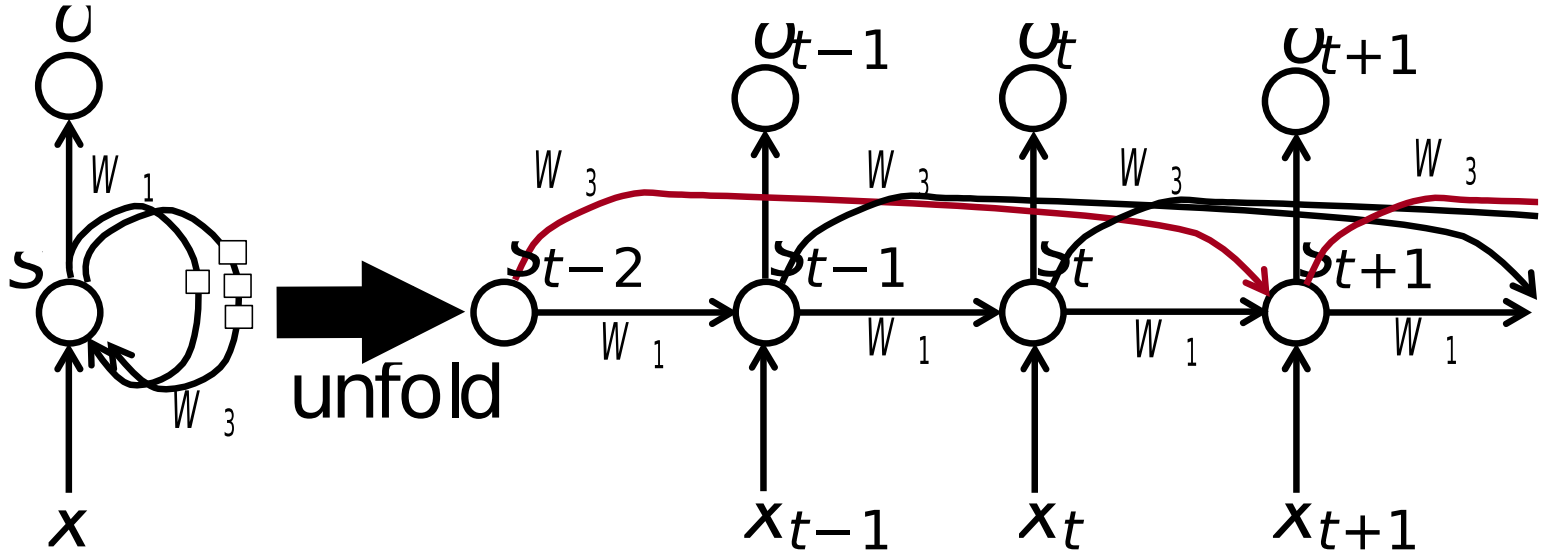


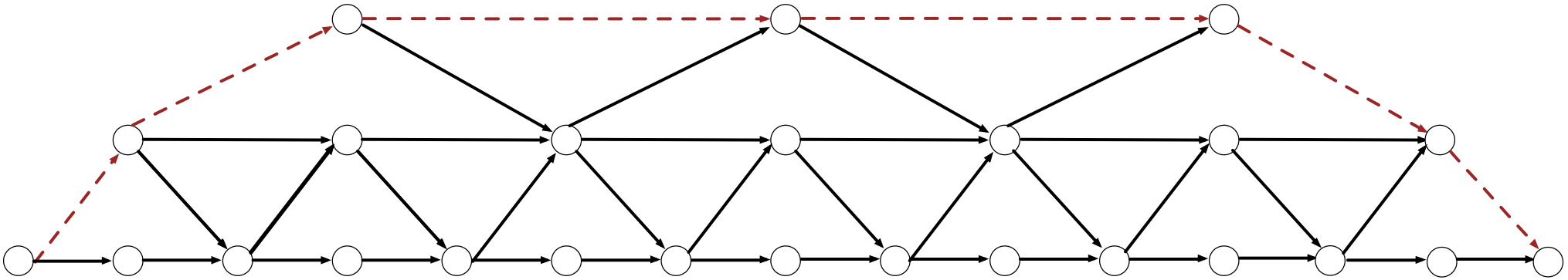


+ skip connections for creating shorter paths

RNN Tricks

- Delays and multiple time scales, Elhihi & Bengio NIPS 1996





Gradient Norm Clipping

(Mikolov thesis 2012;
Pascanu, Mikolov, Bengio, ICML 2013)

$$\hat{\mathbf{g}} \leftarrow \frac{\partial error}{\partial \theta}$$

if $\|\hat{\mathbf{g}}\| \geq threshold$ **then**

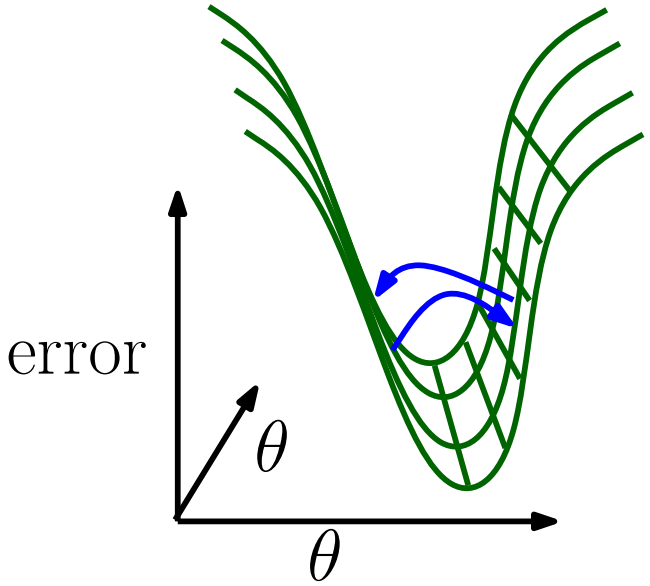
$$\hat{\mathbf{g}} \leftarrow \frac{threshold}{\|\hat{\mathbf{g}}\|}\hat{\mathbf{g}}$$

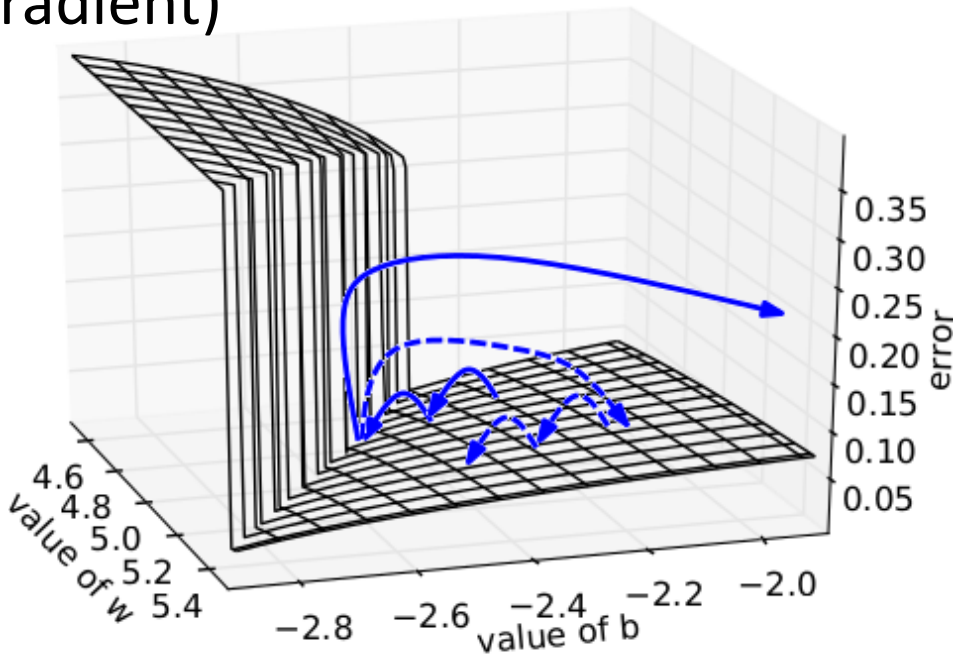
end if

RNN Tricks

(Pascanu, Mikolov, Bengio, ICML 2013; Bengio, Boulanger & Pascanu, ICASSP 2013)

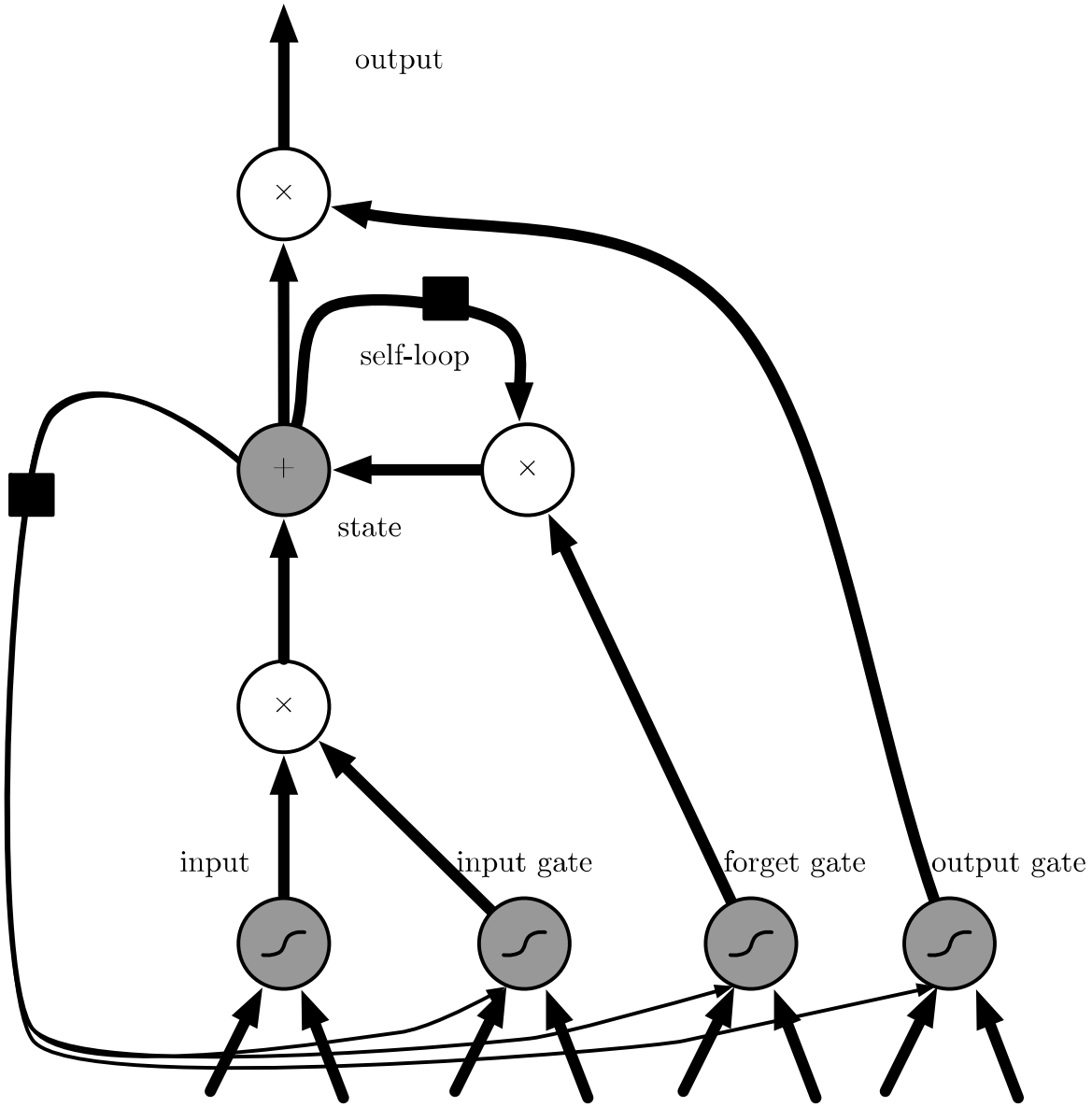
- Clipping gradients (avoid exploding gradients)
- Leaky integration (propagate long-term dependencies)
- Momentum (cheap 2nd order)
- Initialization (start in right ballpark avoids exploding/vanishing)
- Sparse Gradients (symmetry breaking)
- Gradient propagation regularizer (avoid vanishing gradient)
- LSTM self-loops (avoid vanishing gradient)





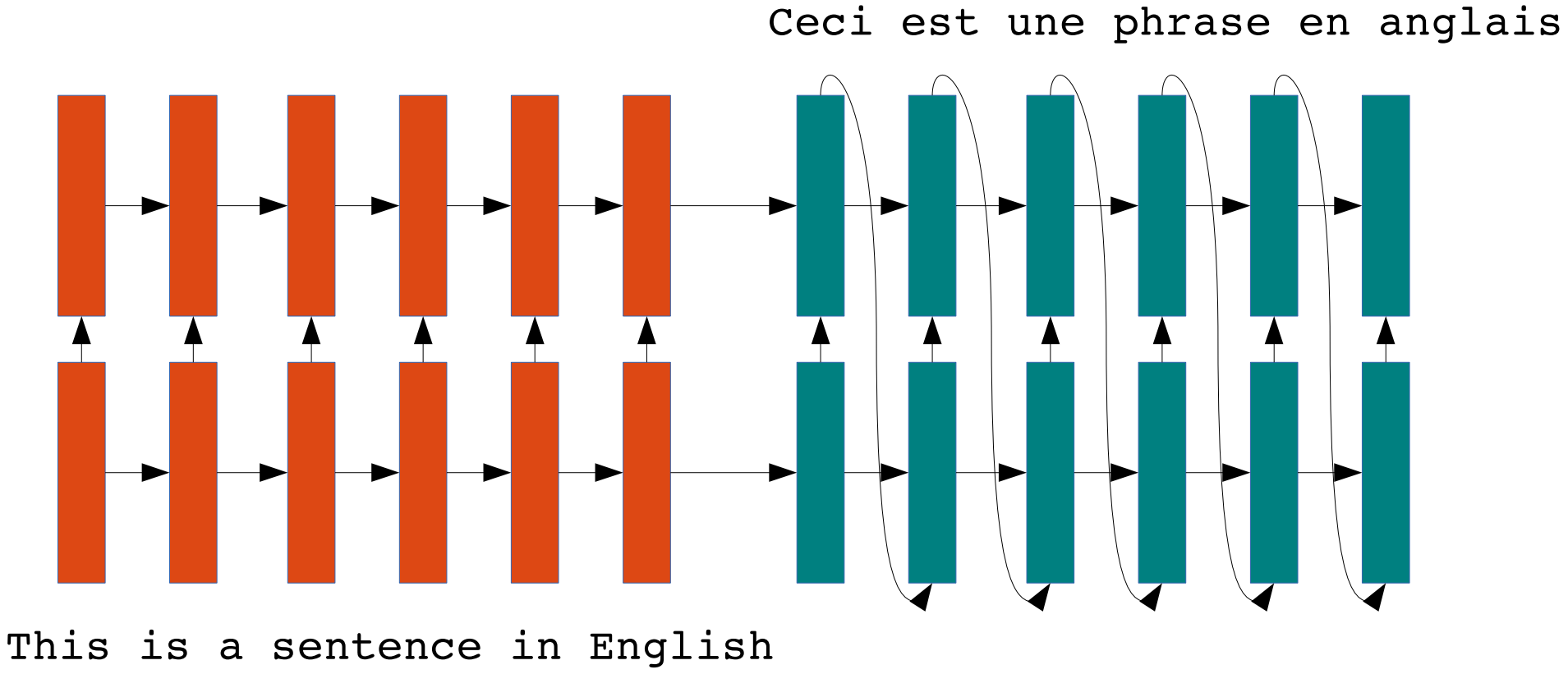
Gated Recurrent Units & LSTM

- Create a path where gradients can flow for longer with self-loop
- Corresponds to an eigenvalue of Jacobian slightly less than 1
- LSTM is **heavily used** *(Hochreiter & Schmidhuber 1997)*
- GRU light-weight version *(Cho et al 2014)*



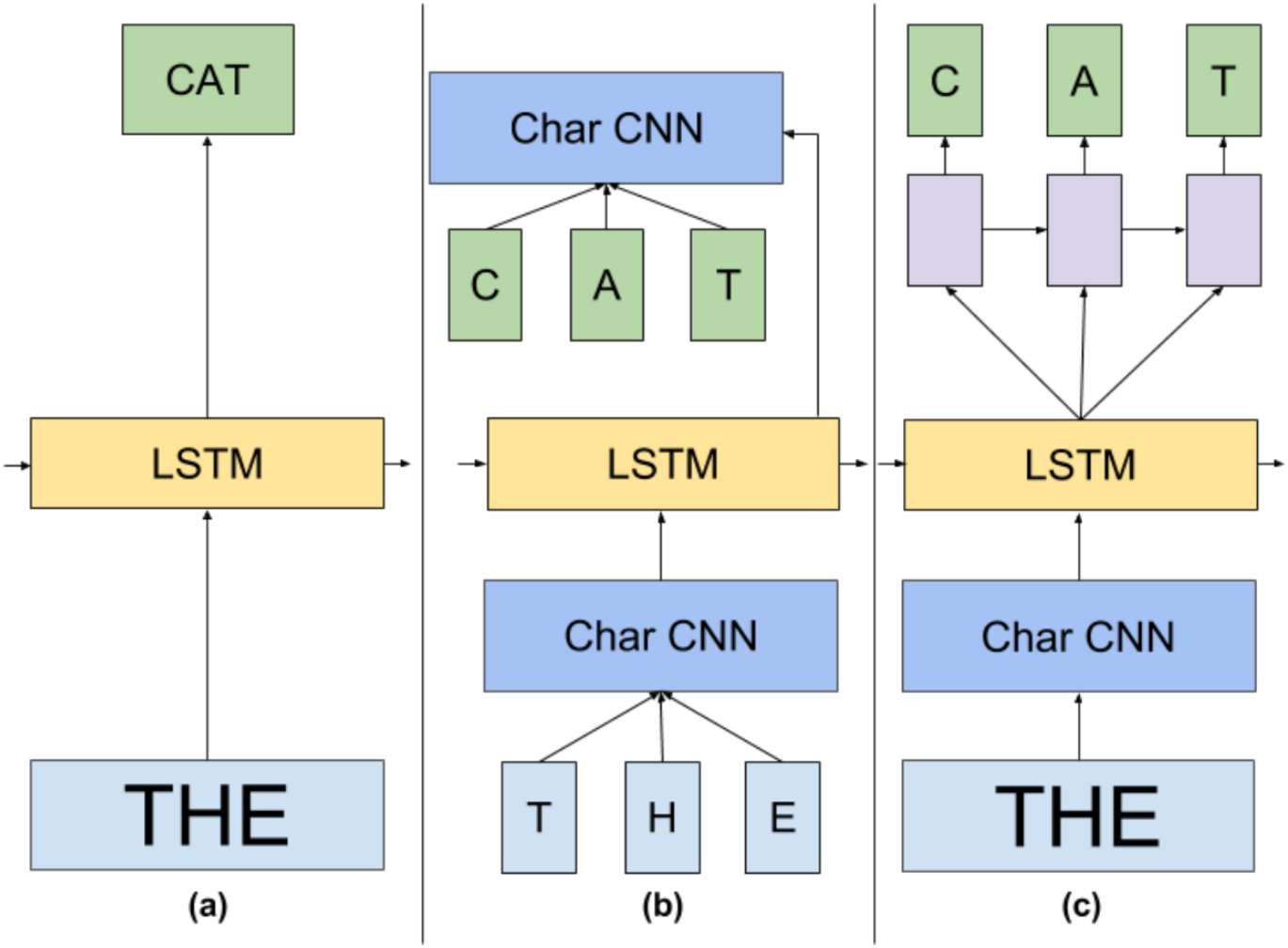
Language Translation with multilayer LSTM networks

- [Sutskever et al. NIPS 2014]
  - Multiple layers of very large LSTM recurrent modules
  - English sentence is read in and encoded
  - French sentence is produced after the end of the English sentence
  - Recent versions (with tricks) have won translation competitions.



LSTM is used in many applications

- Language modeling, speech recognition, translation
  - "Exploring the Limits of Language Modeling" Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, Yonghui Wu. Arxiv:1602.02410



Orthogonal RNN

Make the network quasi-invertible

- "Unitary evolution recurrent neural networks [Arjovsky, Shah, Bengio, arXiv:1511.06464]
- "Recurrent Orthogonal Networks and Long-Memory Tasks" [Henaff, Szlam, LeCun ICML 2016]

- Orthogonal matrix:
  - Satisfy the constraint W.W' = I
- Unitary matrix (matrix with complex numbers):
  - satisfy U.U* = I (U* : transpose and complex conjugate)

Recursive ConvNets

Make the longest path O(log(steps)) instead O(steps)

- A tree of convolutions with shared weights
- Variable number of layers, depending on the size of the input.
  - Until the output has size 1.
- Number of layers = log(size of input)
  - Maximum path length between input and output = log(length of input)